SCHEDULER : A DECISION SUPPORT SYSTEM FOR MACHINE SCHEDULING

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

BABY, T. N.

to the

INDUSTRIAL AND MANAGEMENT ENGINEERING PROGRAMME
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
FEBRUARY, 1986

CERTIFICATE

Certified that the work presented in this thesis entitled 'SCHEDULER: A Decision Support System for Machine Scheduling' by Baby T.N., has been carried out under my supervision and has not been submitted elsewhere for a degree.

February, 1986

(S. Sadagopan)
Assistant Professor
Industrial and Management Engineering Programme,
I.I.T. Kanpur.

ACKNOWLEDGEMENTS

I feel deeply indebted to Dr. S. Sadagopan for his dynamic inspiration and guidance provided throughout the course of preparation of this thesis.

My sincere thanks are due to all my classmates and other friends, who made my stay at I.I.T., Kanpur extremely enjoyable.

Last, in no way the least, I express my thanks to Mr. C.M. Abraham and Mr. Budhi Ram Kandiyal for their excellent work of typing and cyclostyling.

BABY T.N.

CONTENTS

Page

LIST OF FIGURES

ABSTRACT

This thesis develops a simple Decision Support system for Machine Scheduling, which aims at the allocation of available machines over time to best satisfy a set of criteria. Purely manual scheduling systems,which rely on the expertise of a few experienced schedulers do not guarantee optimality. Operations Research/Management Science techniques, which are highly applicable in practical situations are not being fully utilized due to the actual user's not being familiar with them. Computer based decision support systems are found to be very effective in this type of situations. Some of the most popular techniques of scheduling have been incorporated to this system. The algorithms/heuristics include Single or Multiple Machine Scheduling, Job Shop Scheduling and Flow Shop Scheduling.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The role of computers in management has been changing rapidly in the past few years. Starting from simple office automation - wordprocessor and pocket calculators - mechanization of managerial work has now reached a stage where even decision making - the single most important function of a manager - can be assisted directly by a computer. This is the result of our ability to use the computers for jobs which need skills and not merely for routine work.

A significant part of an individual's working and personal time is spent recording/searching for absorbing information. As much as 80 percent of a typical executive's time is spent in processing and communication of information. Computers have become an essential part of organizational information processing because of the power of the technology and the volume of the data to be processed. The application of computers to information processing began in 1954 when one of the first computers was programmed to process payroll. Today computerized processing of transaction data is routine activity of large organizations [13]. Moreover, the capability to automate information processing has permitted an expansion in the scope of

formalized organizational information use. The current challenge in information processing is to use the capabilities of computer to support knowledgework, including managerial activities and decision making. The wide variety of computer resources to perform transaction processing, to provide processing for a formal information and reporting system, and to accomplish managerial-decision support are broadly classified as the organization's 'Management Information System' (MIS).

Information systems provide support for management at all levels: operational control, management control and strategic planning. Each of these class of management activities include, planning, control and decision making. Computer based support systems for these three activities are generally known as 'Decision Support Systems' (DSS). This term, DSS refers to a class of systems which support the process of making decisions. The emphasis is on 'support' rather than an automation of decisions. DSS allow the decision maker to retrieve data and test alternative solutions during the process of problem solving.

Machine scheduling is one area of operations management, where the actual decisions are mostly taken by people at or below the shop supervisor level. This makes it difficult to utilize the OR/MS techniques of optimizing the schedule. For

purely intuitive approaches, optimality is not guaranteed. Also it is quite time consuming and difficult if the jobs need complicated precedence relationships. All these factors point towards the need for a simple interactive system for scheduling, which can use the OR/MS techniques without bothering the user about the theoretical complexity of scheduling problem. It should be able to provide some good reports of alternative schedules possible, a choice for the user to adjust it to his subjective needs and above all a reasonably good initial solution to start with. This thesis tries to make such a system. Though it is not expected to have the finish of a professional package, we have tried to make it functionally satisfactory for the needs described above.

1.2 ORGANIZATION OF THE THESIS

The second chapter describes the evolution and use of DSS and the scheduling problem. The design of the system and techniques used are described in Chapter three. A documentation of the system is necessary for any software package developed. Chapter four is devoted to this purpose. Chapter five, the concluding chapter, gives the conclusions and possible extensions to the system.

A sample problem and its solution through the system is also given as an Appendix.

CHAPTER 2

SCHEDULING AND DSS

2.1 INTRODUCTION

The process of decision making involves the following four phases.

1) Exploration : This is the phase where initial structuring of the situation is accomplished.
2) Modelling and optimization.
3) Interpretation and Postmodelling : Here, the results of the model are translated in plain English and communicated to the decision maker.
4) Implementation.

A lot of progress has been made in Step 2 with the evolution of OR/MS. But the other steps were not seriously looked into until late seventies. This was the big handicap in using OR/MS techniques in solving practical problems effectively as pointed out by authors like Franz Edelman [14] who pinpointed this shortcoming of OR/MS. The evolution of DSS helped to bridge this gap. This chapter describes briefly about DSS and its evolution.

2.2 DECISION MAKING

The efficiency and effectiveness of a decisions are to be viewed distinctly [27]. Efficiency of a decision involves narrowing of focus and minimization of time, cost and effort required to implement the decision. These are generally very much programmatic. This part of the work can be replaced by machine. But effectiveness involves identifying what should be done and ensuring that the chosen criteria is a relevant one. Because this part of the decision making involves of lot of subjectivity it is not programmable. This is the place where a decision maker needs real help. In the case of increasing the efficiency of a decision, OR/MS does prove very helpful through its techniques of optimization. But to incorporate the elements of subjectivity while using these techniques, the decision maker should have an interactive environment for decision making which can handle the complexity of the techniques.

Also the problems faced by decision maker can be classified as structured and unstructured [28]. If a problem is fully structured the decision making process can be done by a machine, whereas in the case of unstructured problems the human element is unavoidable. But a fully structured problem is almost absent in the domain of managerial problems.

Structured and unstructured problems are basically different in another respect also. That is, a structured problem always have, 'the optimum solution', though it may not be practically acceptable to the manager. This makes the decision maker's choice simpler, by giving him a basis to compare with. All the alternative solutions generated are compared with this optimum and the one least deviating from it is selected. This is not so simple in the case of an unstructured problem. There we never have'the optimum solution' What we can do is, always generate more than one alternative solution and compare among them. The one selected can always be improved. Of course, a balance will have to be struck between the improvement and the cost of achieving it.

Because of this difference the number of solutions to be generated and compared will always be more in the case of an unstructured problem. That means an extensive sensitivity analysis which involves a lot of repetitive calculations. Here a computer is the right equipment to use. And the environment which connects the OR/MS techniques and the user and helps him to try out alternatives and choose one through the process described above is called a Decision Support System.

2.3 THE EVOLUTION OF DSS

The concept of decision support evolved from two main areas of research : the theoretical studies of organizational decision making done at Carnegie Institute of Technology during the late 1950s and early '60s and the technical work on interactive computer systems, mainly carried out at Massachusetts Institute of Technology in the 1960s. The Carnegie School provided some key concepts for what H.A. Simon termed as 'new science of management decision' [34]. It emphasized bounded rationality in the individual's decision processes, with the implication that extending the limits on the bounds could improve effectiveness.

Simon's [34] distinction between programmable and non-programmable tasks is central to the argument that Decision Support provides a strategy for making the computer useful to managers, whose decisions are relatively nonstructured. Newell, Simon and Shaw's work [35] on human problem solving has become the most well-established 'behavioural' perspective in MIS and management science. But the technology it implies did not exist until in the 1960s prototype interactive systems were developed. Much of this work was purely technical; M.I.T's project MAC, for example aimed at providing a general-purpose, time-shared system that could be applied in a variety of contexts and would permit 'machine aided cognition'. In 1978 many

of the interactive tools initiated by MAC were becoming common place. It is worth stressing, however, that most of these techniques were developed well before 1970 and they were mainly the outcome of academic and technical research projects. Scott Morton's 'Management Decision Systems' (1971) is the first explicit meshing of the two streams of analysis of decision making and interactive computing. Although his book presented a major technical innovation in organizational use of computers it differed from the dominant focus of the computer scientists - the dynamics of decision making and the organizational implications of changes in information-processing technology.

Between 1971 and 1976 a lot of work on DSS was done. Simultaneously the area of Interactive Computing also developed enough to be used together. There is now an established tradition of work on Decision Support systems and widespread application of them inorganizations. So we can be more confident in our claims that a clear understanding of decision making can provide a framework for assessing and using any new technology. In short, we can claim that Decision Support is a distinctive concept and methodology for developing computer based decision aids.

2.4 ADVANTAGES OF HAVING A DSS

A DSS is very much different from EDP. The latter

replaces only mechanical work whereas the form assists in a mental exercise. A Decision Support System supports and not replaces the managerial judgement. It helps to improve the effectiveness of the decision rather than its efficiency [27]. It increases the economic value of managerial decision by [39].

1. decreasing the cost and time required to perform various phases of decision making,
2. increasing the applicability and efficiency of the process of structuring managerial situations,
3. improving the process of collaboration between manager, OR/MS and information systems analyst.

According to Alter [1] DSS helps in improving consistency and accuracy in problem solving and facility to do experimentation.

Above all it acts as a tool for training and helps the users in learning how to use techniques effectively.

2.5 THE SCHEDULING PROBLEM

2.5.1 Classification :

Machine scheduling can be defined as the allocation of available machines over time to best satisfy some set of criteria. Typically a scheduling problem involves a number of tasks or jobs to be performed. Each of them may have to go through one or more of the machines. There may also be a precedence rule on the operations.

The scheduling problems can be classified in different ways. One of the most popular way of classification is the one described by Graves [18] which is adopted here also. It is based on the processing complexity. Processing complexity is concerned primarily with the number of processing steps associated with each production task or item. This classification is as follows :

- one stage, one processor
- one stage, parallel processors
- multistage , multiprocessors

The one stage, one processor problem is the simplest form of scheduling problem, which is also known as one machine problem. As simple as it is, the single machine problem is very important in learning process of scheduling and its performance measures and also in solving bottleneck situations in large problems. The one stage parallel processor problem is similar to the one machine problem except that each task requires a single step which can be performed on any of the parallel processors. Here more than the scheduling of jobs allocation of resources also becomes relevant as different from single machine case. Also it can be used as a subproblem for job shop case where alternate paths are available for the jobs. The multistage problems is the most general scheduling problem. It can again be sub-divided into flow shop problem - where all the

jobs follow the same precedence rule - and job shop problem - where each job may have different precedence relationships.

2.5.2 Performance Measures for Scheduling

The performance of the schedule may be measured in many ways. Common measures are the utilization level of resources, the percentage of late tasks, average or maximum tardiness for a set of tasks and the average or maximum flow time for a set of tasks. Tardiness is the positive part of the difference between a task's actual completion time and its desired completion time. The flow time for a task is the difference between the completion time of the task and the time at which that task was released to the production shop. In the SCHEDULER the following measures of performance are used :

1) Maximum flow time (F_{max})
2) Mean flow time ($\overline{F}$)
3) Maximum tardiness (T_{max})
4) Mean tardiness ($\overline{T}$)
5) Number of tardijobs (N_T)
6) Percentage utilization of the machine.

A more detailed description on these performance measures and their relevance can be obtained from the book on scheduling by Kenneth. R. Baker [4].

2.5.3 A Review of Scheduling Literature

Production scheduling is also classified as open shop and closed shop problems. Open shop scheduling problem, also called the job shop scheduling problem, may be defined as having to sequence a family of processors so as to complete a given set of tasks and optimize some performance measure. These problems are all combinatorial problems of varying difficulty, and they may all be solved, in theory, by an enumeration strategy such as branch and bound procedure. The closed shop scheduling problem is to find a production schedule to satisfy some given requirements at minimum production cost, where the production schedule specifies both the run quantities for a set of items and the set up sequence for a set of facilities. Our work concentrates mainly on open shop scheduling problems. A brief review of the literature on solution methodologies for these type of problem is given below.

The single machine problem has been the most popular of all scheduling problems. The best known of these results are the procedures for minimizing the mean flow time (Smith, 1956) and for minimizing the maximum tardiness (Jackson, 1955). Both of them achieves the result by simple ordering of the tasks. Moore (1968) developed a slightly more complex procedure for minimizing number of late tasks. The problem of weighted

tardiness was more complex. Karp (1972), Lawler (1964), Emmons (1969), Shwimer (1972), Srinivasan (1971) and Baker and Scharge (1978) have all studied one machine tardiness problem. The Dynamic Programming approach of Scharge and Baker (1978) seems to have substantially tamed this problem.

The one stage- parallel processors problem is an important generalization of single machine problem. Single machine problem is mainly of theoretical interest whereas this is a more practical problem. McNaughton (1959) developed a weighted flow time minimization procedure. When the processors are not identical, Horn (1973) shows how to solve the mean flow time problem. Eastman et al (1964), Rothkopf (1966) and Baker and Mertan (1973) also proposed an tested several heuristics. Hu (1961) and Coffman and Graham (1972) proposed similar listing procedures for non-preemptive cases. For minimizing weighted tardiness algorithms were developed by Root (1965), Elmagharaby (1974) and Barnes and Brennan (1977). Parker et al (1977) used the similarity between this problem and vehicle routing problem to develop heuristics.

For flow shop problems the best known result is Johnson's rule (1954) for two machines. Numerous combinatorial optimization procedures have been proposed for solving general flow shop problem with maximum flow time criterion, e.g., Ashour (1970), Gupta (1971),Ignall and Scharge(1965), McMahon and Burton (1967),

Szwarc (1971). Noteworthy heuristics are the ones by Campbell et al. (1970) and Dannenbring (1977). Job shop problems are the most general and most difficult of all scheduling problems. Most researchers have assumed non-preemptive jobs and maximum flow time as the criterion of optimization. Also most of the algorithms are branch and bound based, e.g., Ashour et al (1974), Balas (1969), Brooks and White (1965), Fisher (1973), Greenberg (1968), and Scharge (1970). There have been two distinct heuristic approaches to the deterministic static job shop problem by Jeremiah (1964) and Gere (1966).

CHAPTER 3

DESIGN OF THE SYSTEM : SCHEDULER

3.1 INTRODUCTION

Before doing the design of any system it is necessary to define the objective of the proposed system. Here in the design of SCHEDULER the main thrust is to allow the user who does not know much of scheduling theory, to use them effectively, not only efficiently. We stress on supporting rather than on replacing the decision. Again we focus on improving the effectiveness of decision making rather than on merely improving its efficiency. These points are talked about in detail in the previous chapter. In decision support we seek to increase decision maker's ability to deal with complexity and uncertainity. It should be a link between the techniques and the applications.

3.2 MOTIVATION FOR DEVELOPING THE SCHEDULER

The first point of motivation is the predominance of purely manual scheduling systems, especially in relatively simple production environments involving only a few processing steps. These systems rely on the expertise of a few experienced schedulers who construct, revise and maintain the production schedule using no more than a few graphical aids such as Gantt chart. To an observer, atleast, it is not often clea

how exactly a schedule is constructed nor alternative schedules are compared or evaluated. The schedule evaluation appears to be qualitative and to be dependent on many criteria. The dominant schedule criteria is often the feasibility, although many other criteria such as flexibility may be important. Nevertheless, such systems seem to work in that the generated schedules are viewed as being quite satisfactory. This, however, is difficult to ascertain due to the qualitative nature of the criteria [18].

In an implementation of a shop floor control system, we are required to select local despatch rule for sequencing tasks waiting at each production processor. Common despatch rules are to sequence the tasks according to expected processing times, slack times or the ratio of task's remaining processing time to the slack time. Shop floor control systems are widely implemented and are credited with producing significant cost savings and performance improvements.

These observations suggests that the current operations research techniques for scheduling may be either mismatched, inadequate or not needed for many production settings. This is only partially true. As Graves [18] points out there is anything but a one to one correspondence between scheduling theory and practice. Scheduling theory is many times too complex for an actual user to use. Still there are encouraging

signs of production settings in which current operations research methodology was proved useful. The discrepancies between theory and practice are yet to be removed. Given the importance of shop floor control systems the necessity of an information and support system is quite obvious. This should be able to fill the gap between the theory and practice to certain extent atleast. Thus the main objective in the design of SCHEDULER is to allow the actual user to use some systematic techniques of scheduling rather than to code a number of complicated mathematical algorithms.

3.3 CONSIDERATIONS IN THE DESIGN OF SCHEDULER

The real problem in evolving a practical computer system for the machine shop was not the question of establishing the right theoretical foundation but rather the need to provide a practical solution to a real world problem. That is, to be acceptable to the actual user, where other than the optimization of performance parameters, a number of subjective and non-quantifiable requirements are to be satisfied. This can be done only by allowing the uses to take part in the actual decision. So SCHEDULER is to be used interactively. As mentioned earlier, since the user develops the actual schedule through a continuous process of changing the initial schedule in steps, evaluating at each time, we have tried to simulate the same in our system.

3.4 ASSUMPTIONS AND LIMITATIONS

The following assumptions are taken in the design of the system :

1. The smallest unit of time used is minute.
2. No job goes through the same machine more than once.
3. When there is a precedence relationship for jobs, no alternative paths are available for any job.
4. No preemption allowed.
5. When the lot size is more than one, the whole lot is considered as one unit for processing.

<u>Limitations :</u> Since most of the algorithms used are heuristics, it may not give an exact optimum solution. It is justified by the fact that, the initial solution any way after adjusting for user's subjective needs will be away from optimum.

Some of the above said assumptions limits the scope of the SCHEDULER, especially the last one.

3.5 ALGORITHMS/HEURISTICS USED IN THE SCHEDULER

3.5.1 Single Machine Problem

a) SPT ordering : This optimizes mean flow time, number of elements in the system and the mean waiting time. It is only the ordering of the jobs in non-decreasing order of processing time.

b) EDD sequencing : In EDD sequencing we arrange the jobs in increasing order of due dates. It minimizes maximum tardiness.

c) Hodgson's algorithm : Here the objective is to minimize the number of tardy jobs. This algorithm uses a modified EDD sequencing to achieve the objective.

d) Wilkerson-Irwin algorithm : Here mathematically developed conditions are used as the despatch rule. Objective is to minimize the mean tardiness.

e) Dynamic programming : This again optimizes the mean tardiness. It involves more computation than the Wilkelson-Irwin algorithm but gives a better solution. The approach is to do an implicit enumeration of all schedules. It applies the principle of optimality to curtail the process of enumeration.

3.5.2 Parallel Machine Problem

In the case of parallel machine problem makespan is the usual optimization criterion. Here we have used two algorithms for minimizing makespan and mean flow time.

1. McNaughton's Algorithm : This minimizes the makespan for m parallel, identical machine case where preemption is allowed. Though our systems does not allow preemption this becomes useful when lot size is more than one.

2. Heuristic-1 : Here again we minimizes makespan but preemption is not allowed. This basically uses LPT ordering to achieve the optimum. This any way does not guarantee an optimum.
3. Heuristic-2 : Mean flow time is minimized in this heuristic. SPT ordering is used in this procedure to get a minimum makespan.

3.5.3 The m-Machine, n-Job Problem

In m-machine, n-job problems makespan is the most common measure of performance. So the algorithms developed are also generally aimed at minimizing makespan.

1. Flow Shop (Johnson's rule) : This is used to solve flow shop problems of 2 or 3 machines. It optimizes makespan.
2. Job Shop Scheduling (Heuristic procedure) : This heuristic uses the principle of active schedule generation. A set of despatching rules developed by Jeremiah, Lalchandani and Scharge [24] are used to schedule the jobs. In this implementation the rule used is the most work remaining. We select the operation associated with the job having the most work remaining to be processed.

Algorithms like WSPT are not implemented separately because the processing times used in calculation of schedules are adjusted for their weightages, if any, at the beginning itself.

CHAPTER 4

THE SCHEDULER - IMPLEMENTATION

4.1 INTRODUCTION

The SCHEDULER is intended to assist the user in scheduling the m number of machines for n number of jobs. The range of problems which can be solved using the SCHEDULER varies from the simplest single machine problem to the most general job shop problem of m-machines and n jobs with each job having a different precedence relationship. The problem can have some unstructured elements which may not necessarily fit in to the input model of any of the algorithms used in the system. In such cases what we do is as follows. The problem is approximated to suit the algorithms available and solved. As many solutions as the number of algorithms are generated. Many times there may be repetition of schedules. These redundancies are removed through a simple check. Now the resulting schedules are adjusted to fit the unstructured constraints. This is the first output of the system.

Now, according to the user's request changes are made in the schedule and other elements affecting the schedule . After accepting and effecting these changes in the first output schedule they are again displayed back to the user with the new values of performance parameters calculated. In case the user

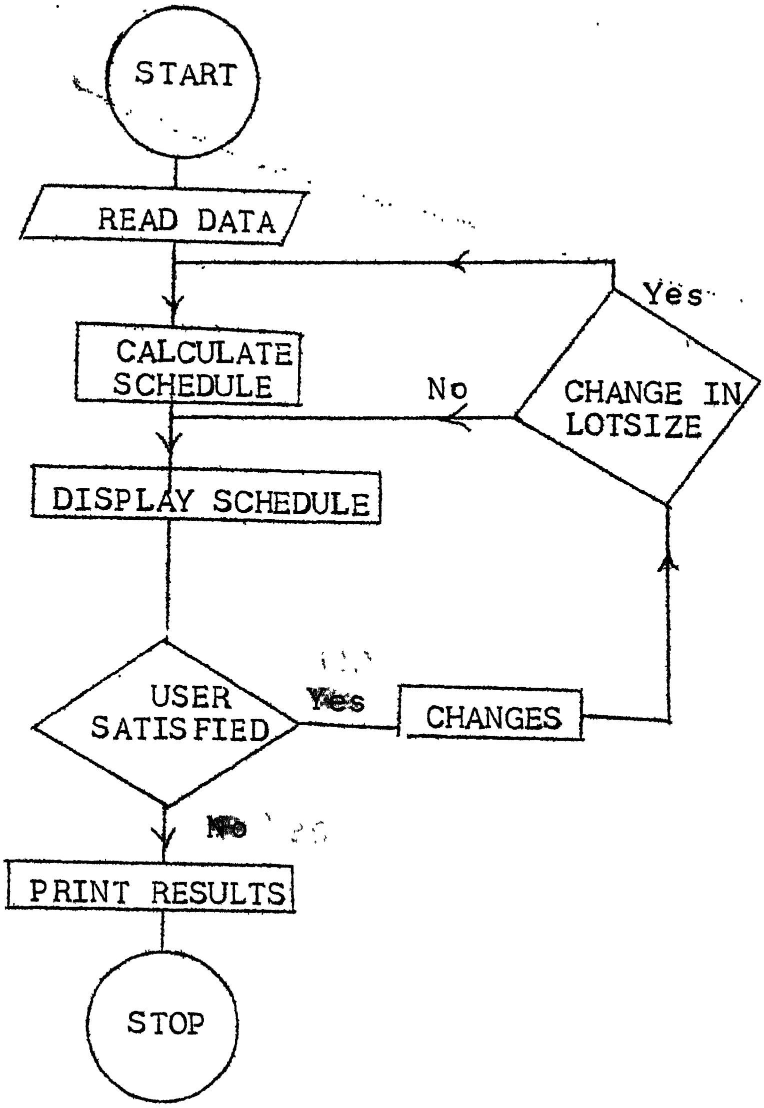


Fig. 4.1 The SCHEDULER

is interested in changing lot sizes also, schedules may be again calculated by using the appropriate algorithms. The flow diagram in Fig. 4.1 gives an overview of the working of the system.

The SCHEDULER is programmed in PASCAL. It contains approximately 2500 lines in more than forty procedures. The following sections describe the major subsystems of the SCHEDULER and brief descriptions of important procedures.

4.2 DATA COLLECTION SUBSYSTEM

This part, through a simple question answer session acquires enough data for the purpose of scheduling. The data include :

1) number of machines (m)
2) number of jobs (n)
3) precedence relationships of each job, if any
4) the processing time of each job-i on each machine -j(t_{ij})
5) The due date of each job-i (d_i)
6) weightages (w_i), lot sizes (t_i) and set up times (ST_i)
7) temporary unavailability of machines, if any.

The procedures used in this part of the program are listed below :

READDATA : This in turn calls the procedures PRECEDENCE PROCESSING-TIME, LOTSIZE, WEIGHTAGE, SETUPTIME, UNAVAILABLE-TIME and MAKE-TIME.

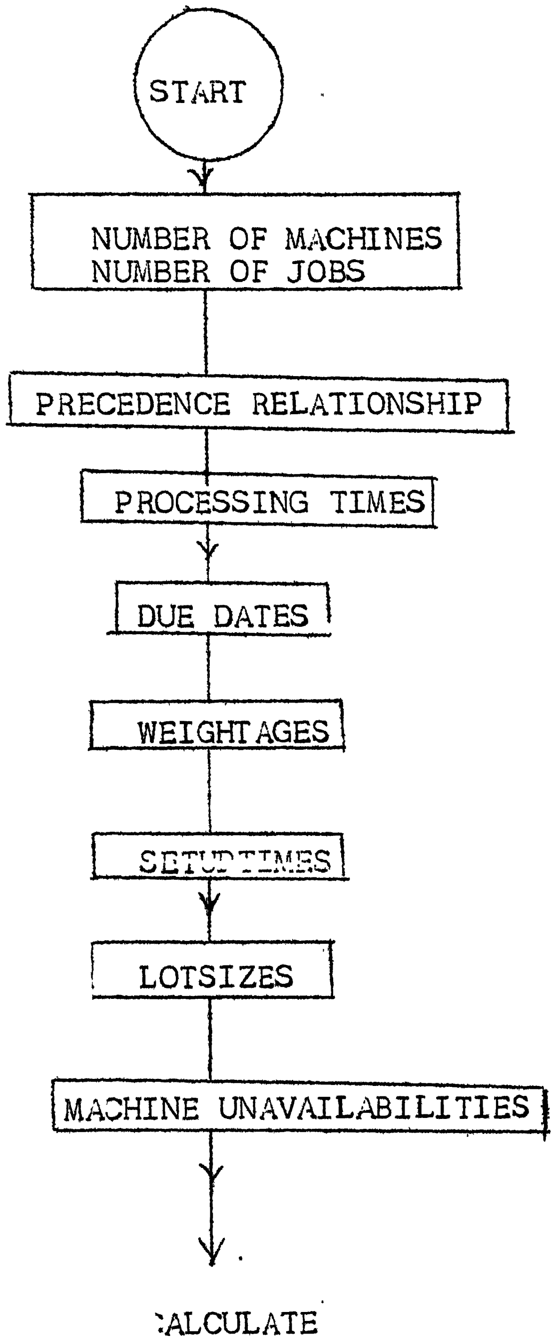


Fig. 4.2 READDATA

The output of the procedures as obvious from the name, will be the problem specifications stored in data structures suitable for transfer to other procedures.

PRECEDENCE : This gets the number of jobs and machines and the precedence-relations of jobs, if any. Also this procedure decide model to be selected.

PROCESSING-TIME : The function of this procedure is to collect the processing time of all the operations.

DUE-DATE : The due dates of all the jobs are recorded here. Also the starting time and date of the schedule is also collected here.

WEIGHTAGE : The part of input collected by this procedure is the relative weightages of the job, if any, for the user.

SETUPTIME : In case the machines have some setuptime this procedure rewill record them.

LOTSIZE : When the job is produced in lots the lot sizes need to be known. This procedure takes them as input from user.

UNAVAILABLE-TIME : Some times the machines may have some temporary unavailability. This information is obtained from the user by this procedure.

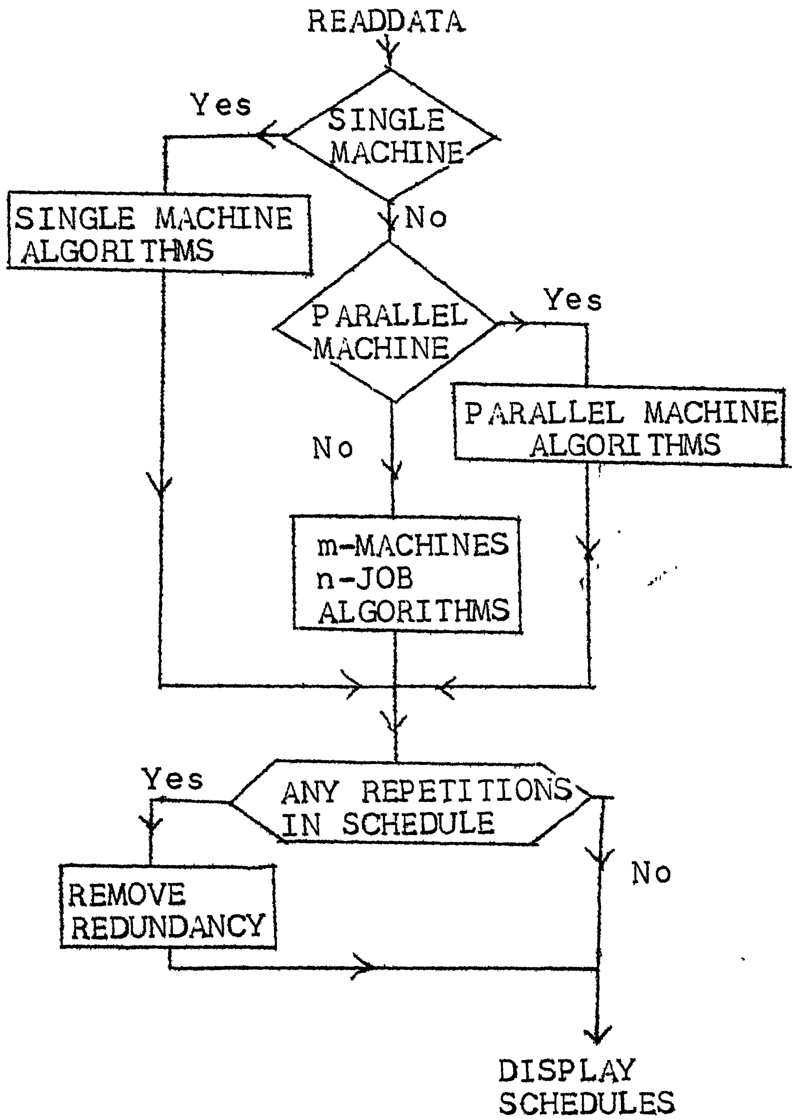


Fig. 4.3 CALCULATE

MAKE-TIME : All the above said data need to be put in proper form to be used by other procedures. The conversion takes place in this procedure.

4.3 SCHEDULE CALCULATION SUBSYSTEM

Computing the schedule : This step involves the use of some algorithm appropriate for the problem and calculating the schedule when there are more than one algorithm for the same model, this system proceeds the following way. All algorithms are used to find as many schedules as possible. Then if any redundancy of schedules exists, they are eliminated. At this stage we use the following procedures to calculate the schedule.

CALCULATE : This procedure depending on the model selected calls the Procedures SOLVE1 (for single m/c case), SOLVE2 (for parallel m/c case) or SOLVE3 (for mxn case).

SOLVE1 : This computes the schedules for single machine problem using a number of algorithms.

SOLVE2 : To calculate the schedules for parallel machine case we use this procedure.

SOLVE3 : This is the procedure which calls procedures for algorithms in case the problem is a m-machine, n-job type.

Details of the algorithms used are described in Section 3.5.

4.4 OUTPUT SUBSYSTEM

This part of the system gives the user the output of the systems in the form of charts and tables. The major outputs include the schedule and its performance measures. The schedule is displayed in the form of a Gantt chart. After displaying the performance measures, at the user's request, details such as starting and finishing time of each operation, finishing time and tardiness of each job, percentage utilization of each machine etc. are also displayed. These results can be supplied in the form of a disk file also if the user wishes so.

The start-finish times and performance measures of the jobs are calculated by the following procedures.

ONEMCSF : This procedure is used exclusively for single machine problem. A separate procedure is written for single machine case to reduce the memory requirement.

PARALLELMCSF : For parallel machine problem as the number of jobs are different on each machine, this separate procedure was required.

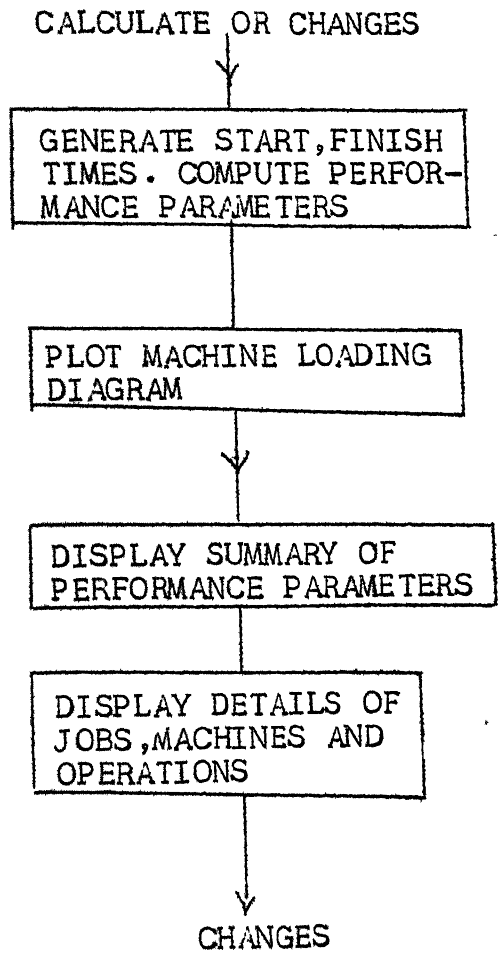


Fig. 4.4 DISPLAY SCHEDULES

MMCNJSF : For a m-machine, n-job problem to find start, finish times and to calculate the performance measures this procedure is used.

The machine loading diagram is displayed using a graphics package IGL (PLOT-10). Since access to IGL is easier through FORTRAN the corresponding subroutine was written in FORTRAN. This introduces a little extra effort in programming because a linking of PASCAL-FORTRAN I/O was necessary. This difficulty is overcome using a linking procedure 'FTNLNK' available in the DEC-10 installation on which the SCHEDULER is implemented. The following are the other procedures used in the output subsystem.

REPORT : This procedure calls the appropriate procedure for finding start, finish times and calculating the performance measures.

EXHIBIT-SINGLE : This procedure is used when the user is interested in seeing any single schedule only.

4.5 MODIFICATION SUBSYSTEM

The next step in the working of the system is to accept the changes in the schedule displayed in Step 3. Facility is provided to change not only schedule but also the temporary unavailability of the machines. This will help the user to schedule the time of his planned maintenance.

In case the user is interested in changing the lot sizes - to find the optimum lot size so that the machine idle times will be reduced - a facility for that also is provided. In this case the schedule will be again calculated by going through step 2. Otherwise we will go back to step 3 (see Fig. 4.1) and display the new schedule incorporating new changes. The process of going through steps 2,3 and 4 is continued until the user gets a satisfactory schedule. This is the step where the user actually manipulates the schedule and takes part in the decision, thus utilizing the power of a DSS. The only procedure used in this part of the system is CHANGES, which accepts the changes and passes back the control to the appropriate step.

Flow diagrams for subsystems are also shown in Figures 4.2 to 4.5.

Some special case problems can be solved by using the SCHEDULER intelligently. Two of those are described below.

1. Suppose a job does not go through all machines. This can be taken care of by assuming the processing time of the job on that machine as zero.
2. In job shop problems,if a job has got alternative paths the problem can be solved in two steps. The first step being the decision about which path to follow and the second phase, the usual scheduling problem. The first phase can be modelled as a parallel machine problem. Once

the decision on path is taken the problem becomes easier and can be solved as usual.

A sample run of the SCHEDULER is given in Appendix.

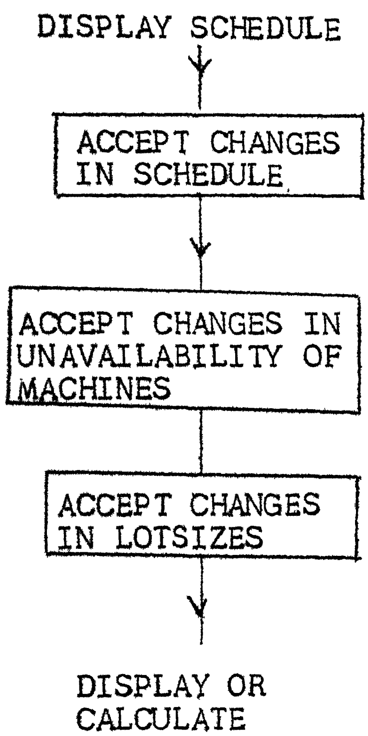


Fig. 4.5 CHANGES

CHAPTER 5

CONCLUSION

The work reported in this thesis is a modest attempt to develop a simple decision support system for machine scheduling. The necessary software for the purpose is developed and implemented on the DEC-1090 computer at I.I.T., Kanpur. Though the work is not expected to be a professional software package, attempts to incorporate all the important characteristics of a Decision Support system have been made. It is intended to support an actual practitioner of scheduling. Scheduling being a decision taken at the shop floor level, simplicity is the usage of the system is attributed as much importance as the optimality of the solutions generated. A modular approach is used in the development of the software, so that addition of new algorithms/heuristics and report generators can be done very easily.

Success of a system such as SCHEDULER depends to a large extent on its acceptability by the intended user. Experimentation of SCHEDULER on real life situations would be of immense value in testing its acceptance. However, so far it has been tested only in an academic setting. The limited experience has been encouraging.

Recommendations

The SCHEDULER does not take care of complicated situations like sequence dependent set up times, alternate paths for jobs, etc. These can be taken care of when a better system is developed. Also closed shop scheduling problems, where lot sizing also will have to be done, can be included. A good interaction between the actual users of the system and the designers of the system is necessary throughout the development of the system. After the development the users' reactions and comments should be used to update the system periodically. Maintenance and updating of a DSS is as important as its development.

REFERENCES

1. Alter, Steven, L., (1980), Decision Support Systems - Current Practices and Continuing Challenges, Addison-Wesly Series on Decision Support.

2. Ashour, S. (1970), A branch and bound algorithm for flow shop scheduling problems, AIIE Trans., vol. 2, 172-176.

3. Ashour, S., T.E. Moore, and K.Y. Chiu, (1974), An implicit enumeration algorithm for the nonpreemptive shop scheduling problem, AIIE Trans., vol.6, 62-72.

4. Baker, K.R. (1974), Introduction to Sequencing and Scheduling, John Wiley and Sons, New York.

5. Baker, K.R. (1975), A comparative study of flow shop algorithms, Operations Research, vol. 23, 62-73.

6. Baker, K.R., and A.G. Merten (1973), Scheduling with parallel processors and linear delay costs, Naval Res. Logist. Quart., vol. 20, 193-204.

7. Baker, K.R., and L. Scharge,(1978), Finding optimal sequence by dynamic programming : An extension to precedence related tasks, Operations Research, vol. 26 , 111-120.

8. Balas, E. (1969), Machine sequencing via disjunctive graphs: An implicit enumeration algorithm, Operations Research, vol. 17, 941-957.

9. Barnes, J.W., and J.J. Brennan, (1977), An improved algorithm for scheduling jobs on identical machines, AIIE Trans., vol.9, 25-31.

10. Brooks, G.H., and C.R. White, (1965), An algorithm for finding optimal or near optimal solutions to the production scheduling problem, Journal of Industrial Engg. vol. 16, 34-40.

11. Campbell, H.G., R.A. Dudek and M.L. Smith (1970), A heuristic algorithm for the n-job, m-machine sequencing problem, Mgt. Sci., vol. 16, 630-637.

12. Dannenbring, D.G. (1977), An evolution of flow shop sequencing heuristics, Mgt. Sci., vol. 23, 1174-1182.

13. Davis, G.B. and M.H. Olson, (1985), Management Informatic… Systems, McGraw Hill Publication.

14. Edelman, Franz (1977), They went thataway, Interfaces, vol. 7, 39-43.

15. Elmaghraby, S.E., and S. Park (1974), On the scheduling of jobs on a number of identical machines, AIIE Trans. vol. 6, 1-13.

16. Emmons, H. (1969), One machine sequencing to minimize certain functions of job tardiness, Operations Research. vol. 17, 701-715.

17. Fisher, M.L. (1973), Optimal solution of scheduling problems using Lagrange multipliers - Part I, Operati… Research, vol. 21, 1114-1127.

18. Graves, S.C. (1981), A review of production scheduli… Operations Research, vol. 29, 646-675.

19. Gupta, J.N.D. (1971), An improved algorithm for the fl… shop scheduling problem, Operations Research, vol. 19, 1753-1758.

20. Horn, W.A. (1973), Minimizing average flow time with parallel machines, Operations Research, vol. 21, 846-8…

21. Hu, T.C. (1961), Parallel sequencing and assembly line problems, Operations Research, vol. 9, 841-848.

22. Ignall, E., and L. Scharge (1965), Application of the branch and bound technique to some flow shop scheduling problems, Operations Research, vol. 13, 400-412.

23. Jackson, J.R. (1955), Scheduling a production line to minimize maximum tardiness, Research Report 43, Managem… Sciences Research Project, UCLA.

24. Jeremiah, B., A. Lalchandani and L. Scharge, (1964), Heuristic rules towards optimal scheduling, Research R… Department of Industrial Engineering, Cornell Univer…

25. Johnson, S.M. (1954), Optimal two and three stage pr… ction schedules with setuptimes included, Naval Res. Logist. Quart., vol. 1, 61-68.

26. Karp, R.M. (1972), Reducibility among combinatorial problems,In complexity of computer applications, 85-103, Plenum Press, NY.

27. Keen, P.G.W., and M.S.S. Morton (1978), Decision Support Systems:an organizational perspective, Addision Wesly Series on Decision Support.

28. Lawler, E.L. (1964), On scheduling problems with deffered deferred costs, Mgt. Sci., vol. 11, 280-288.

29. McMahon, G.B., and P.G. Burton, (1967), Flow shop scheduling with the Branch and Bound method, Operations Research, vol. 15, 473-481.

30. McNaughton, R. (1959), Scheduling with dead lines and loss functions, Mgt. Sci., vol. 6, 1-12.

31. Moore, J.M. (1968), An N job machine sequencing algorithm for minimizing the number of late jobs, Mgt. Sci., vol.15, 102-109.

32. Parker, R.G., R.H. Deane and R.A. Holmes, (1977), On the use of a vehicle routing algorithm for the parallel processor problems with sequence dependent changeover costs, AIIE Trans., vol.9, 155-160.

33. Root, J.G. (1965), Scheduling with deadlines and loss functions on K parallel machines, Mgt. Sci., vol. 11, 460-475.

34. Rothkopf, M. (1966), Scheduling independent tasks on parallel processors, Mgt. Sci., vol. 12, 437-477.

35. Scharge, L., (1970), Solving resource constrained network problems by implicit enumeration - non preemptive cases, Operations Research, vol. 18, 263-278.

36. Shwimer, J., (1972), On the n-job one-machine, sequence dependent problem with tardiness penalties: a Branch and Bound approach, Mgt. Sci., vol. 18, B301-B313.

37. Simon, H.A., (1965), The new science of management decision, Harper and Row, NY.

38. Simon, H.A., A. Newett, (1958), Heuristic problem solving: The next advance in operations research, Operations Research, vol.6, 1-10.

39. Srinivasan, V., (1971), A hybrid algorithm for the one-machine sequencing problem to minimize total tardiness, Naval. Res. Logist. Quart., vol. 18, 317-327.

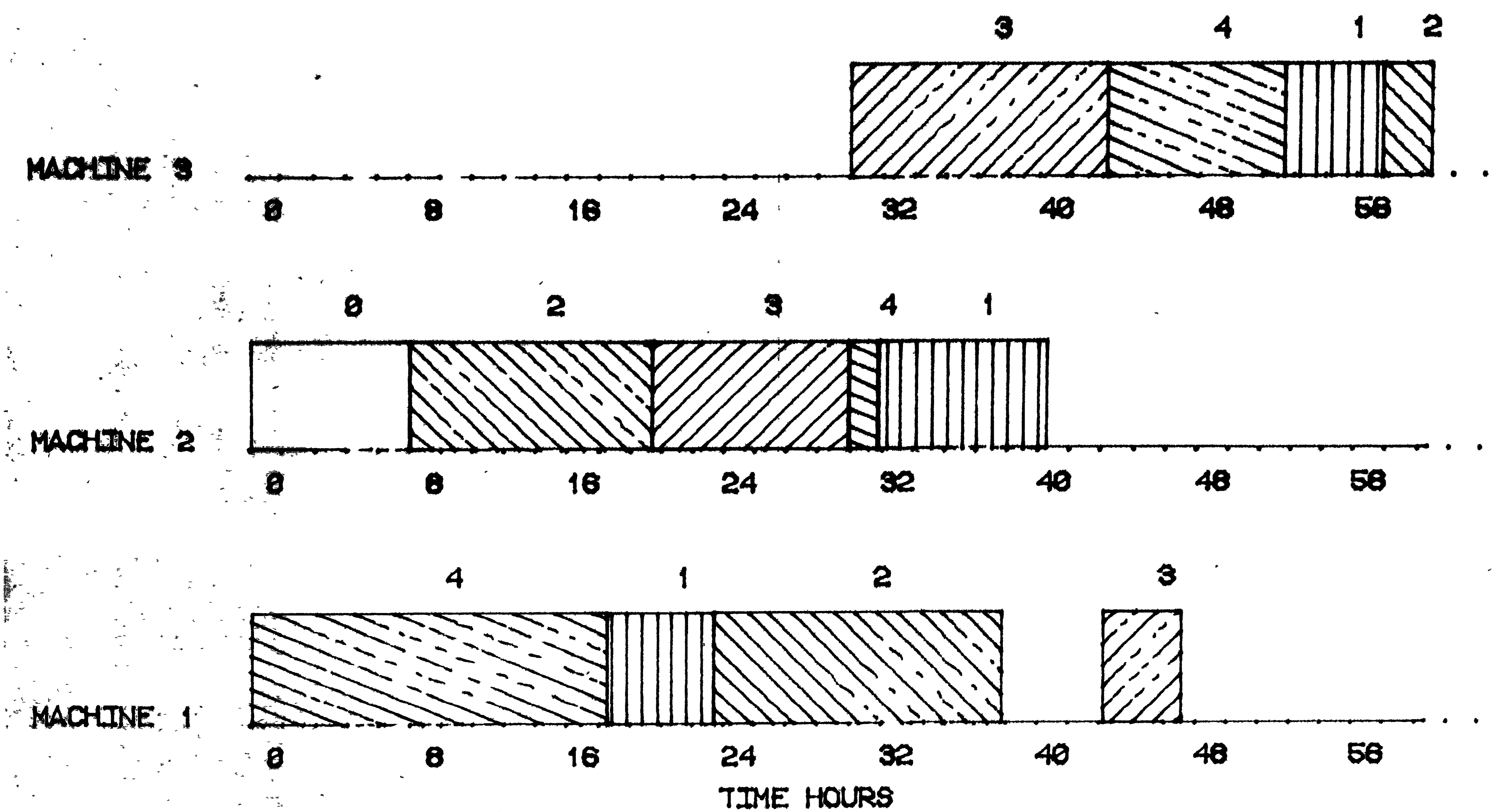
40. Szwarc, W. (1971), Elimination methods in the mxn sequencing problem, Naval. Res. Quart., vol.18, 295-305.

41. Vazsonyi, Andrew, (1978), Decision support systems: The new technology of decision making?, Interfaces, vol.9, 72-77.

SAMPLE SESSION - I

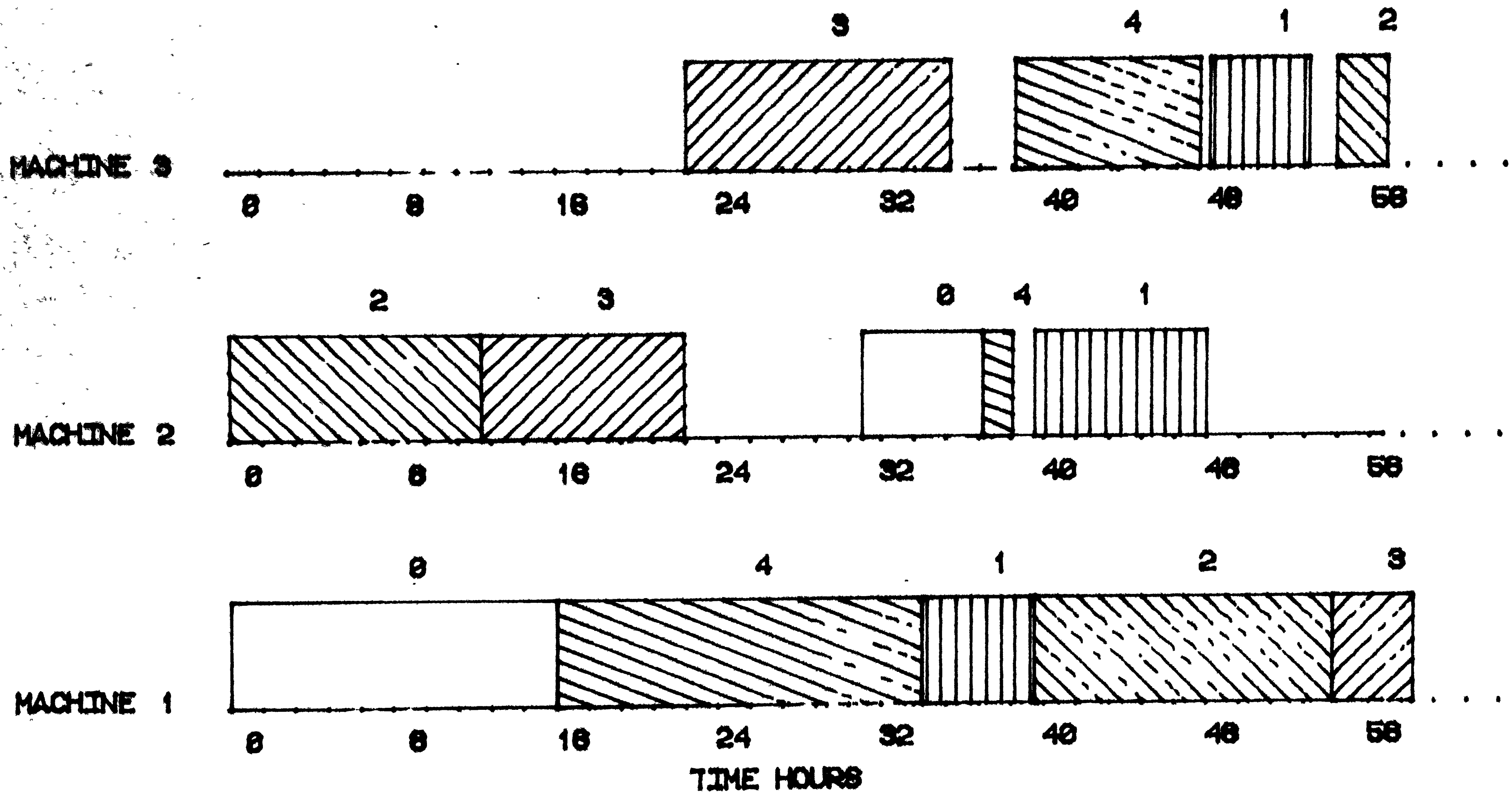
```
.EX T.PAS,T1.FOR,SYS:FTNLNK.REL,/SEA SYS:IGL.REL
[8:55:53]
LINK:   Loading
%LNKMDS Multiply-defined global symbol FORER.
        Detected in module FORINI from file SYS:FORLIB.REL[1,4]
        Defined value = 510650, this value = 56504
[LNKXCT T execution]

*

   PLEASE TYPE IN THE NO. OF MACHINES : 3
   PLEASE TYPE IN THE NO. OF JOBS : 4
   HOW MANY EIGHT HOUR SHIFTS DO YOU HAVE IN A DAY ?  : 2
   GIVE STARTING TIME OF THE FIRST SHIFT : 8
ARE ALL MACHINES IDENTICAL(Y/N)? : N
 ARE ALL JOBS FOLLOWING SAME PATH(Y/N)? : N
 GIVE PRECEDENCE RELATIONSHIP FOR JOB  1  : 1 2 3
 GIVE PRECEDENCE RELATIONSHIP FOR JOB  2  : 2 1 3
 GIVE PRECEDENCE RELATIONSHIP FOR JOB  3  : 2 3 1
 GIVE PRECEDENCE RELATIONSHIP FOR JOB  4  : 1 2 3
ARE ALL OPERATIONS HAVING SAME TIME(Y/N)? : N
 GIVE PROCESSING TIME OF JOB  1  :
 ON MACHINE  1 (DAYS HOURS MINUTES)  : 0 3 30
 ON MACHINE  2 (DAYS HOURS MINUTES)  : 0 7 30
 ON MACHINE  3 (DAYS HOURS MINUTES)  : 0 4 25
 GIVE PROCESSING TIME OF JOB  2  :
 ON MACHINE  1 (DAYS HOURS MINUTES)  : 0 6 20
 ON MACHINE  2 (DAYS HOURS MINUTES)  : 0 5 40
 ON MACHINE  3 (DAYS HOURS MINUTES)  : 0 1 00
 GIVE PROCESSING TIME OF JOB  3  :
 ON MACHINE  1 (DAYS HOURS MINUTES)  : 0 2 00
 ON MACHINE  2 (DAYS HOURS MINUTES)  : 0 9 00
 ON MACHINE  3 (DAYS HOURS MINUTES)  : 0 12 30
 GIVE PROCESSING TIME OF JOB  4  :
 ON MACHINE  1 (DAYS HOURS MINUTES)  : 1 0 0
 ON MACHINE  2 (DAYS HOURS MINUTES)  : 0 0 30
 ON MACHINE  3 (DAYS HOURS MINUTES)  : 0 8 30
 WHEN DO YOU THINK THE PROCESSING WILL START? :
 STARTING DATE (yymmdd) : 860201
 STARTING TIME (HOUR MINUTE) : 8 00
 ARE ALL DUE DATES SAME(Y/N)? : N
 GIVE DUE DATE OF JOB  1  :
 DATE (yymmdd) : 860202
 HOUR MINUTE : 8 00
 GIVE DUE DATE OF JOB  2  :
 DATE (yymmdd) : 860201
 HOUR MINUTE : 20 00
 GIVE DUE DATE OF JOB  3  :
 DATE (yymmdd) : 860210
 HOUR MINUTE : 8 0
 GIVE DUE DATE OF JOB  4  :
 DATE (yymmdd) : 860203
 HOUR MINUTE : 12 0
   DO YOU WANT TO GIVE DIFFERENT WEIGHTAGES TO THE JOBS ? (Y/N)   : Y
   GIVE WEIGHT FOR
   JOB   1 :   1
   JOB   2 :   1
   JOB   3 :   3
   JOB   4 :   1
```

```
ARE THE JOBS BEING PRODUCED IN LOTS OF SIZE MORE THEAN ONE ? (Y/N)    : Y
 ARE ALL THE LOTSIZES OF THE SAME SIZE ?  (Y/N)    : N
 GIVE LOTSIZE OF
   JOB   1 :   1
   JOB   2 :   2
   JOB   3 :   1
   JOB   4 :   1
  ARE THE MACHINES HAVING A SETUP TIME ? (Y/N)   : Y
  ARE ALL  MACHINES HAVING EQUAL SETUP TIME ? (Y/N)  : N
    GIVE SETUP TIME OF
   MACHINE  1 (MINUTES) :   120
   MACHINE  2 (MINUTES) :   60
   MACHINE  3 (MINUTES) :   30
  ANY TEMPORARY UNAVAILABILITY FOR ANY MACHINES ? (Y/N)  : Y
  IS IT THERE FOR
  MACHINE  1 ?  : N
  MACHINE  2 ?  : Y
  IS IT ROUTINE OR CASUAL ? (C/R) :  C
  GIVE DATE (yymmdd) :  860201
  FROM WHAT TIME ? (HOUR MINUTE  ) : 8 0
  UPTO WHAT TIME ? (HOUR MINUTE  ) : 17 0
  MACHINE  3 ?  : N
  YOU HAVE FOUR OPTIONS IN SEEING THE SHEDUL DETAILS.
              1 . ALL SCHEDULES IN DETAIL AND A SUMMARY AT THE END.
              2 . ONLY SCHEDULES IN DETAIL.
              3 . ONLY SUMMARY .
              4 . ANY SPECIFIC SCHEDULE IN DETAIL.
     WHICH ONE DO YOU WANT ? (TYPE IN THE NO.) : 1
THE MACHINE LOADING DIAGRAM OF THE CURRENT SCHEDULE IS AS FOLLOWS
TYPE  C  TO CONTINUE : C
TYPE  C  TO CONTINUE : C
      MEAN FLOWTIME =  3310 MINUTES
 MAXIMIUM FLOW TIME =  3645 MINUTES
     MEAN TARDINESS =  1625 MINUTES
  MAXIMUM TARDINESS =  2925 MINUTES
  NO. OF TARDY JOBS =     3
DO YOU WANT MORE DETAIL(Y/N)?   : N
  TYPE  C  TO CONTINUE.  : C
SERIAL      ..FLOWTIME..          ..TARDINESS..     NO. OF
 NO.       MAXIMUM     MEAN     MAXIMUM     MEAN   TARDYJOBS
   1        3645       3310      2925       1625       3
         (ALL IN MINUTES)
  IN CASE ANY OF THESE SCHEDULES SHOWN NOW ARE NOT AT ALL OF INTEREST
    TO YOU WE WILL DELETE FROM STORAGE.
  DO YOU FEEL SO ?(Y/N) : N
 DO YOU WANT SEE THE SCHEDULES AGAIN ? (Y/N) : N
 DO YOU WANT TO CHANGE ANY SCHEDUL ? (Y/N) : N
 DO YOU WANT TO CHANGE THE TIME UNAVAILABILITY OF ANY M/C ? (Y/N)   : Y
 FOR WHICH M/C DO YOU WANT TO MAKE CHANGE ?    : 1
 GIVE NEW DATE (yymmdd) :   860201
 GIVE NEW UNAVAILABILITY STARTING TIME (HOUR MINUTE)   : 8 0
 GIVE NEW UNAVAILABILITY ENDING TIME (HOUR MINUTE)   : 24 0
DO YOU WANT TO MAKE CHANGES IN THE LOTSIZE ? (Y/N) :N
  YOU HAVE FOUR OPTIONS IN SEEING THE SHEDUL DETAILS.
              1 . ALL SCHEDULES IN DETAIL AND A SUMMARY AT THE END.
              2 . ONLY SCHEDULES IN DETAIL.
              3 . ONLY SUMMARY .
              4 . ANY SPECIFIC SCHEDULE IN DETAIL.
     WHICH ONE DO YOU WANT ? (TYPE IN THE NO.) : 1
```



MACHINE LOADING DIAGRAM

```
THE MACHINE LOADING DIAGRAM OF THE CURRENT SCHEDULE IS AS FOLLOWS
TYPE  C  TO CONTINUE : C
TYPE  C  TO CONTINUE : C
      MEAN FLOWTIME =  3124 MINUTES
 MAXIMIUM FLOW TIME =  3400 MINUTES
     MEAN TARDINESS =  1351 MINUTES
  MAXIMUM TARDINESS =  2680 MINUTES
  NO. OF TARDY JOBS =     3
DO YOU WANT MORE DETAIL(Y/N)?   : Y
              ..DUE..          ..FINISHING..
JOB NO.    DATE    TIME       DATE       TIME  FLOWTIME TARDINESS
   1     860202     8: 0    860204      12:55   3175    2215
   2     860201    20: 0    860204      16:40   3400    2680
   3     860210     8: 0    860204      14:10   3250       0
   4     860203    12: 0    860203      20:30   2670     510
TYPE  C  TO CONTINUE : C
M/C NO.  TOTAL TIME    IDLE TIME   P.c UTILIZATION
          PER CYCLE     PER CYCLE    of THE M/C
   1          3250          0          100.00
   2          2130         160          92.48
   3          3400        1635          51.91
  TYPE  C  TO CONTINUE.  : C
SERIAL      ..FLOWTIME..          ..TARDINESS..     NO. OF
 NO.       MAXIMUM ,  MEAN     MAXIMUM      MEAN   TARDYJOBS
    1       3400     3124      2680      1351         3
         (ALL IN MINUTES)
  IN CASE ANY OF THESE SCHEDULES SHOWN NOW ARE NOT AT ALL OF INTEREST
    TO YOU WE WILL DELETE FROM STORAGE.
  DO YOU FEEL SO ?(Y/N) : N
 DO YOU WANT SEE THE SCHEDULES AGAIN ? (Y/N) : N
 DO YOU WANT TO CHANGE ANY SCHEDUL ? (Y/N) : N
 DO YOU WANT TO CHANGE THE TIME UNAVAILABILITY OF ANY M/C ? (Y/N)   : Y
 FOR WHICH M/C DO YOU WANT TO MAKE CHANGE ?    : 2
 GIVE NEW DATE (yymmdd) :   860203
 GIVE NEW UNAVAILABILITY STARTING TIME (HOUR MINUTE)    : 8 0
 GIVE NEW UNAVAILABILITY ENDING TIME (HOUR MINUTE)   : 14 0
DO YOU WANT TO MAKE CHANGES IN THE LOTSIZE ? (Y/N) :N
  YOU HAVE FOUR OPTIONS IN SEEING THE SHEDUL DETAILS.
               1 . ALL SCHEDULES IN DETAIL AND A SUMMARY AT THE END.
               2 . ONLY SCHEDULES IN DETAIL.
               3 . ONLY SUMMARY .
               4 . ANY SPECIFIC SCHEDULE IN DETAIL.
     WHICH ONE DO YOU WANT ? (TYPE IN THE NO.) : 1
THE MACHINE LOADING DIAGRAM OF THE CURRENT SCHEDULE IS AS FOLLOWS
TYPE  C  TO CONTINUE : C
TYPE  C  TO CONTINUE : C
      MEAN FLOWTIME =  3191 MINUTES
 MAXIMIUM FLOW TIME =  3400 MINUTES
     MEAN TARDINESS =  1419 MINUTES
  MAXIMUM TARDINESS =  2680 MINUTES
  NO. OF TARDY JOBS =     3
DO YOU WANT MORE DETAIL(Y/N)?   : Y
              ..DUE..          ..FINISHING..
JOB NO.    DATE    TIME       DATE       TIME  FLOWTIME TARDINESS
   1     860202     8: 0    860204      13:25   3205    2245
   2     860201    20: 0    860204      16:40   3400    2680
   3     860210     8: 0    860204      14:10   3250       0
   4     860203    12: 0    860204       8:30   2910     750
TYPE  C  TO CONTINUE : C
M/C NO.  TOTAL TIME    IDLE TIME   P.c UTILIZATION
          PER CYCLE     PER CYCLE    of THE M/C
   1          3250          0          100.00
   2          2370         580          75.52
   3          3400        1635          51.91
```



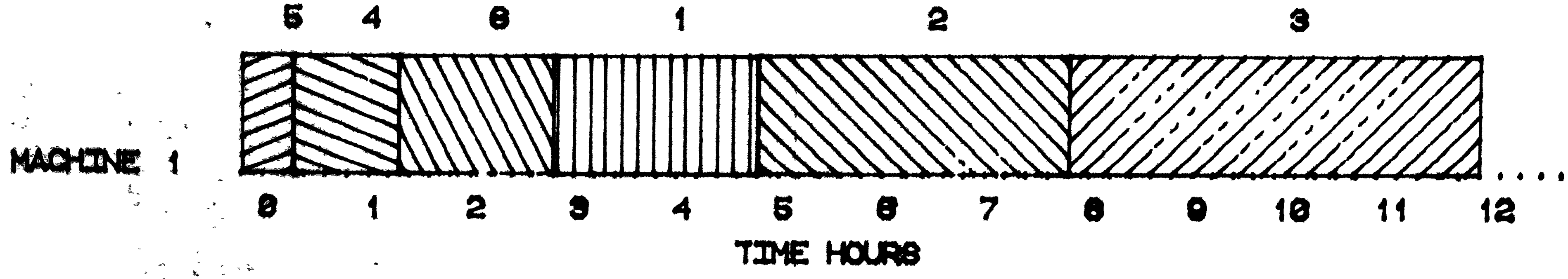
MACHINE LOADING DIAGRAM

```
   TYPE  C  TO CONTINUE.  : C
 SERIAL      ..FLOWTIME..          ..TARDINESS..      NO. OF
  NO.      MAXIMUM      MEAN     MAXIMUM     MEAN   TARDYJOBS
    1       3400       3191      2680      1419         3
          (ALL IN MINUTES)
   IN CASE ANY OF THESE SCHEDULES SHOWN NOW ARE NOT AT ALL OF INTEREST
     TO YOU WE WILL DELETE FROM STORAGE.
   DO YOU FEEL SO ?(Y/N) : N
  DO YOU WANT SEE THE SCHEDULES AGAIN ? (Y/N) : N
  DO YOU WANT TO CHANGE ANY SCHEDUL ? (Y/N) : N
  DO YOU WANT TO CHANGE THE TIME UNAVAILABILITY OF ANY M/C ? (Y/N)   : N
 DO YOU WANT TO MAKE CHANGES IN THE LOTSIZE ? (Y/N) :N
  ANY MORE PROBLEMS ? (Y/N) : N

EXIT
```

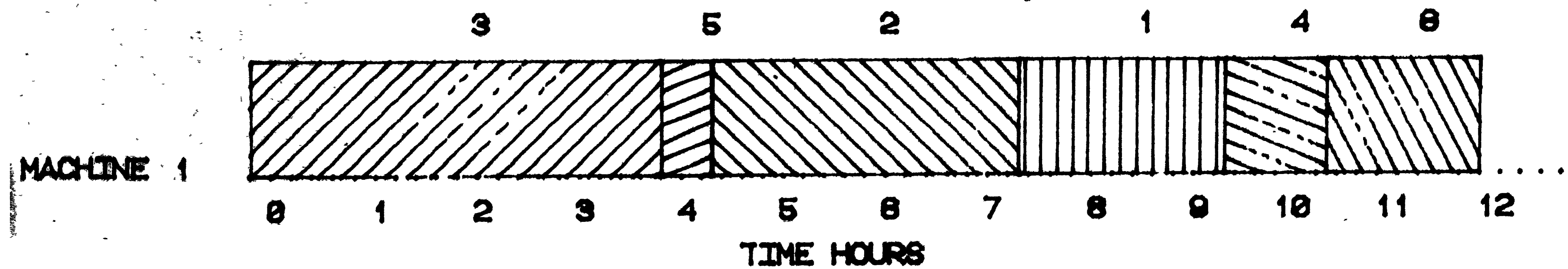
SAMPLE SESSION - II

```
EX T.PAS,T1.FOR,SYS:FTNLNK.REL,/SEA SYS:IOL.REL
 [14:45:38]
 LINK:   Loading
 %LNKMDS Multiply-defined global symbol FORER.
         Detected in module FORINI from file SYS:FORLIB.REL[1,4]
         Defined value = 510647, this value = 56732
 [LNKXCT T execution]

 *

   PLEASE TYPE IN THE NO. OF MACHINES : 1
   PLEASE TYPE IN THE NO. OF JOBS : 6
   HOW MANY EIGHT HOUR SHIFTS DO YOU HAVE IN A DAY ?  : 1
   GIVE STARTING TIME OF THE FIRST SHIFT : 8
 ARE ALL OPERATIONS HAVING SAME TIME(Y/N)? : N
  GIVE PROCESSING TIME OF JOB  1  : 0 2 0
  GIVE PROCESSING TIME OF JOB  2  : 0 3 0
  GIVE PROCESSING TIME OF JOB  3  : 0 4 0
  GIVE PROCESSING TIME OF JOB  4  : 0 1 0
  GIVE PROCESSING TIME OF JOB  5  : Q 0 30
  GIVE PROCESSING TIME OF JOB  6  : 0 1 30
  WHEN DO YOU THINK THE PROCESSING WILL START? :
  STARTING DATE (yymmdd) : 860201
  STARTING TIME (HOUR MINUTE) : 8 0
  ARE ALL DUE DATES SAME(Y/N)? : N
  GIVE DUE DATE OF JOB  1  :
  DATE (yymmdd) : 860202
  HOUR MINUTE : 10 0
  GIVE DUE DATE OF JOB  2  :
  DATE (yymmdd) : 860201
  HOUR MINUTE : 16 0
  GIVE DUE DATE OF JOB  3  :
  DATE (yymmdd) : 860201
  HOUR MINUTE : 12 0
```

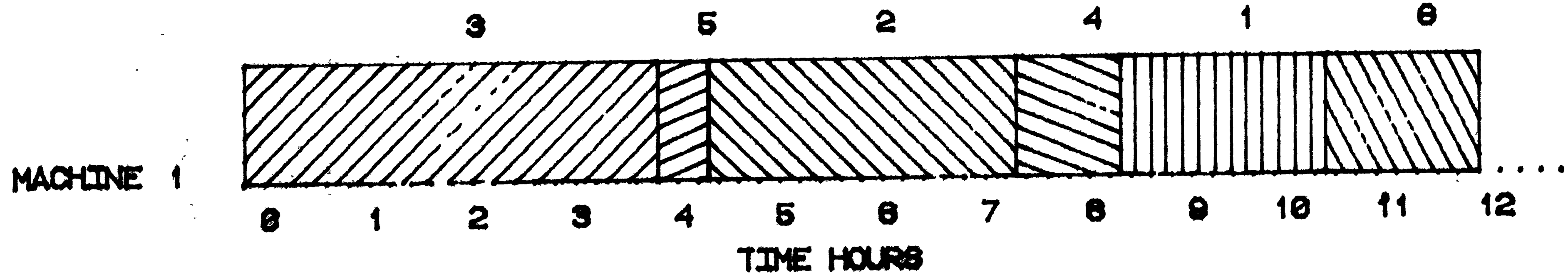
```
GIVE DUE DATE OF JOB  4  :
DATE (yymmdd) : 860202
HOUR MINUTE : 10 0
GIVE DUE DATE OF JOB  5  :
DATE (yymmdd) : 860201
HOUR MINUTE : 14 0
GIVE DUE DATE OF JOB  6  :
DATE (yymmdd) : 860202
HOUR MINUTE : 12 0
  DO YOU WANT TO GIVE DIFFERENT WEIGHTAGES TO THE JOBS ? (Y/N)  : N
 ARE THE JOBS BEING PRODUCED IN LOTS OF SIZE MORE THEAN ONE ? (Y/N)   : N
  ARE THE MACHINES HAVING A SETUP TIME ? (Y/N)   : N
  ANY TEMPORARY UNAVAILABILITY FOR ANY MACHINES ? (Y/N)  : N
  YOU HAVE FOUR OPTIONS IN SEEING THE SHEDUL DETAILS.
             1 . ALL SCHEDULES IN DETAIL AND A SUMMARY AT THE END.
             2 . ONLY SCHEDULES IN DETAIL.
             3 . ONLY SUMMARY .
             4 . ANY SPECIFIC SCHEDULE IN DETAIL.
     WHICH ONE DO YOU WANT ? (TYPE IN THE NO.) : 1
THE MACHINE LOADING DIAGRAM OF THE CURRENT SCHEDULE IS AS FOLLOWS
TYPE  C  TO CONTINUE : C
TYPE  C  TO CONTINUE : C
      MEAN FLOWTIME =   300 MINUTES
 MAXIMIUM FLOW TIME =   720 MINUTES
     MEAN TARDINESS =    20 MINUTES
  MAXIMUM TARDINESS =   120 MINUTES
   NO. OF TARDY JOBS =      1
DO YOU WANT MORE DETAIL(Y/N)?   : Y
              ..DUE..          ..FINISHING..
JOB NO.    DATE     TIME      DATE      TIME  FLOWTIME TARDINESS
   1     860202    10: 0    860201      8:30     30       0
   2     860202     8: 0    860201      9:30     90       0
   3     860201    12: 0    860201     11: 0    180       0
   4     860202    10: 0    860201     13: 0    300       0
   5     860201    14: 0    860202      8: 0    480     120
   6     860202    12: 0    860202     12: 0    720       0
TYPE  C  TO CONTINUE : C
M/C NO.  TOTAL TIME    IDLE TIME   p.c UTILIZATION
          PER CYCLE    PER CYCLE    of THE M/C
   1         720           0          100.00
THE MACHINE LOADING DIAGRAM OF THE CURRENT SCHEDULE IS AS FOLLOWS
TYPE  C  TO CONTINUE : C
TYPE  C  TO CONTINUE : C
      MEAN FLOWTIME =   480 MINUTES
 MAXIMIUM FLOW TIME =   720 MINUTES
     MEAN TARDINESS =    80 MINUTES
  MAXIMUM TARDINESS =   270 MINUTES
   NO. OF TARDY JOBS =      2
DO YOU WANT MORE DETAIL(Y/N)?   : Y
              ..DUE..          ..FINISHING..
JOB NO.    DATE     TIME      DATE      TIME  FLOWTIME TARDINESS
   1     860202    10: 0    860201     12: 0    240       0
   2     860202     8: 0    860201     12:30    270       0
   3     860201    12: 0    860201     15:30    450     210
   4     860202    10: 0    860202      9:30    570       0
   5     860201    14: 0    860202     10:30    630     270
   6     860202    12: 0    860202     12: 0    720       0
TYPE  C  TO CONTINUE : C
M/C NO.  TOTAL TIME    IDLE TIME   p.c UTILIZATION
          PER CYCLE    PER CYCLE    of THE M/C
   1         720           0          100.00
```



ACHINE LOADING DIAGRAM



MACHINE LOADING DIAGRAM



MACHINE LOADING DIAGRAM

```
THE MACHINE LOADING DIAGRAM OF THE CURRENT SCHEDULE IS AS FOLLOWS
TYPE  C  TO CONTINUE : C
TYPE  C  TO CONTINUE : C
      MEAN FLOWTIME =   470 MINUTES
 MAXIMUM FLOW TIME =   720 MINUTES
     MEAN TARDINESS =    80 MINUTES
  MAXIMUM TARDINESS =   270 MINUTES
  NO. OF TARDY JOBS =     2
DO YOU WANT MORE DETAIL(Y/N)?   : Y
              ..DUE..          ..FINISHING..
JOB NO.    DATE     TIME       DATE       TIME  FLOWTIME TARDINESS
   1     860202    10: 0    860201     12: 0     240       0
   2     860202     8: 0    860201     12:30     270       0
   3     860201    12: 0    860201     15:30     450     210
   4     860202    10: 0    860202      8:30     510       0
   5     860201    14: 0    860202     10:30     630     270
   6     860202    12: 0    860202     12: 0     720       0
TYPE  C  TO CONTINUE : C
M/C NO.  TOTAL TIME    IDLE TIME   P.C UTILIZATION
          PER CYCLE     PER CYCLE    of THE M/C
   1         720           0          100.00
THE MACHINE LOADING DIAGRAM OF THE CURRENT SCHEDULE IS AS FOLLOWS
TYPE  C  TO CONTINUE : C
TYPE  C  TO CONTINUE : C
      MEAN FLOWTIME =   470 MINUTES
 MAXIMUM FLOW TIME =   720 MINUTES
     MEAN TARDINESS =    80 MINUTES
  MAXIMUM TARDINESS =   270 MINUTES
  NO. OF TARDY JOBS =     2
DO YOU WANT MORE DETAIL(Y/N)?   : Y
              ..DUE..          ..FINISHING..
JOB NO.    DATE     TIME       DATE       TIME  FLOWTIME TARDINESS
   1     860202    10: 0    860201     12: 0     240       0
   2     860202     8: 0    860201     12:30     270       0
   3     860201    12: 0    860201     15:30     450     210
   4     860202    10: 0    860202      8:30     510       0
   5     860201    14: 0    860202     10:30     630     270
   6     860202    12: 0    860202     12: 0     720       0
TYPE  C  TO CONTINUE : C
M/C NO.  TOTAL TIME    IDLE TIME   P.C UTILIZATION
          PER CYCLE     PER CYCLE    of THE M/C
   1         720           0          100.00
  TYPE  C  TO CONTINUE.  : C
SERIAL       ..FLOWTIME..           ..TARDINESS..      NO. OF
 NO.       MAXIMUM     MEAN      MAXIMUM      MEAN    TARDY JOBS
    1        720        400        120          20         1
    2        720        480        270          80         2
    3        720        470        270          80         2
    4        720        470        270          80         2
        (ALL IN MINUTES)
  IN CASE ANY OF THESE SCHEDULES SHOWN NOW ARE NOT AT ALL OF INTEREST
    TO YOU WE WILL DELETE FROM STORAGE.
  DO YOU FEEL SO ?(Y/N) : N
 DO YOU WANT SEE THE SCHEDULES AGAIN ? (Y/N) : N
 DO YOU WANT TO CHANGE ANY SCHEDULE ? (Y/N) : N
 DO YOU WANT TO CHANGE THE TIME UNAVAILABILITY OF ANY M/C ? (Y/N)   : N
DO YOU WANT TO MAKE CHANGES IN THE LOTSIZE ? (Y/N) :N
 ANY MORE PROBLEMS ? (Y/N) : N
```